

Lista de exercícios Funções - Passagem de parâmetros por referência

1. Faça uma função que receba dois números inteiros **a** e **b**, em seguida, decmente o primeiro e incremente o segundo. No exercício, **a** e **b** devem ser passados por referência. Depois teste, invocando a função criada. Protótipo:

```
void questao01(int *a, int *b)
```

2. Crie uma função que receba os valores **a**, **b** e **c** passados por valor, e receba também dois valores **x1** e **x2** passados por referência. Em seguida, calcule e retorne as duas raízes da equação do segundo grau nas variáveis **x1** e **x2**. Protótipo:

```
int calcula_raizes(float a, float b, float c, float *x1, float *x2)
```

3. Desenvolva um programa que leia a quantidade total de segundos e converta para Horas, Minutos e Segundos. Imprima o resultado da conversão no formato HH:MM:SS. Para isso, utilize a função com protótipo

```
void converteHora(int total segundos, int *hora, int *min, int *seg)
```

4. Faça os seguintes itens:

- (a) Implemente uma função com protótipo:

```
int divide(int *m, int *n, int d);
```

que recebe três parâmetros (dois ponteiros para inteiro e um inteiro) e retorna 1 se **d** divide pelo menos um entre ***m** e ***n**, 0 caso contrário. Fora isso, se **d** divide ***m**, a função deve dividir ***m** por **d**, e o mesmo para ***n**.

- (b) Escreva uma função com protótipo:

```
int mmc (int m, int n);
```

que recebe como parâmetros dois inteiros positivos **m** e **n** e retorna, usando a função `divide (...)` acima, o mínimo múltiplo comum entre **m** e **n**, ou seja, `mmc(m,n)`.

- (c) Escreva um programa que lê **n** inteiros positivos e calcula, usando a função `mmc(...)` acima, o mínimo múltiplo comum entre todos eles.

5. Suponha a seguinte função, a qual retorna o primeiro e o último dígito de um número:

```
void removeExtremos(int *n, int *pri, int *ult){
    int tn,pot = 1;
    tn = *n;
    while(tn >= 10){
        tn = tn/10;
        pot *= 10;
    }
    *pri = *n / pot;
    *ult = *n % 10;
    *n = *n % pot;
    *n = *n / 10;
}
```

Faça um programa que utilize esta função para determinar se um determinado inteiro positivo **n** é palíndromo. Suponha que **n** **não** contém o dígito 0. Atenção: você não deve modificar a função dada no enunciado! Apenas invocá-la de forma adequada.